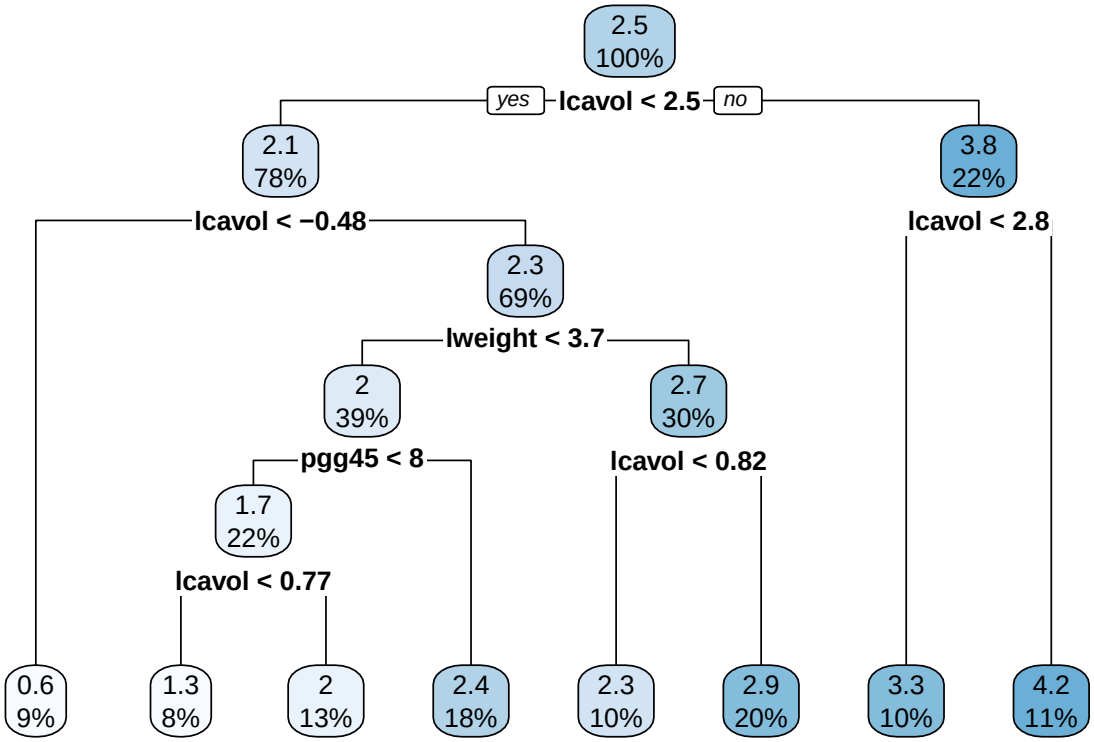
Homework 4

Deepika Dilip

4/19/2019

Problem 1a: Regression Tree

```
#Importing Data
data(Prostate)
prostate <- Prostate

#Regression Tree : Initial (complexity parameter =0.01)
set.seed(2)
tree0 <- rpart(formula = lpsa~., data = prostate)
rpart.plot(tree0)
```

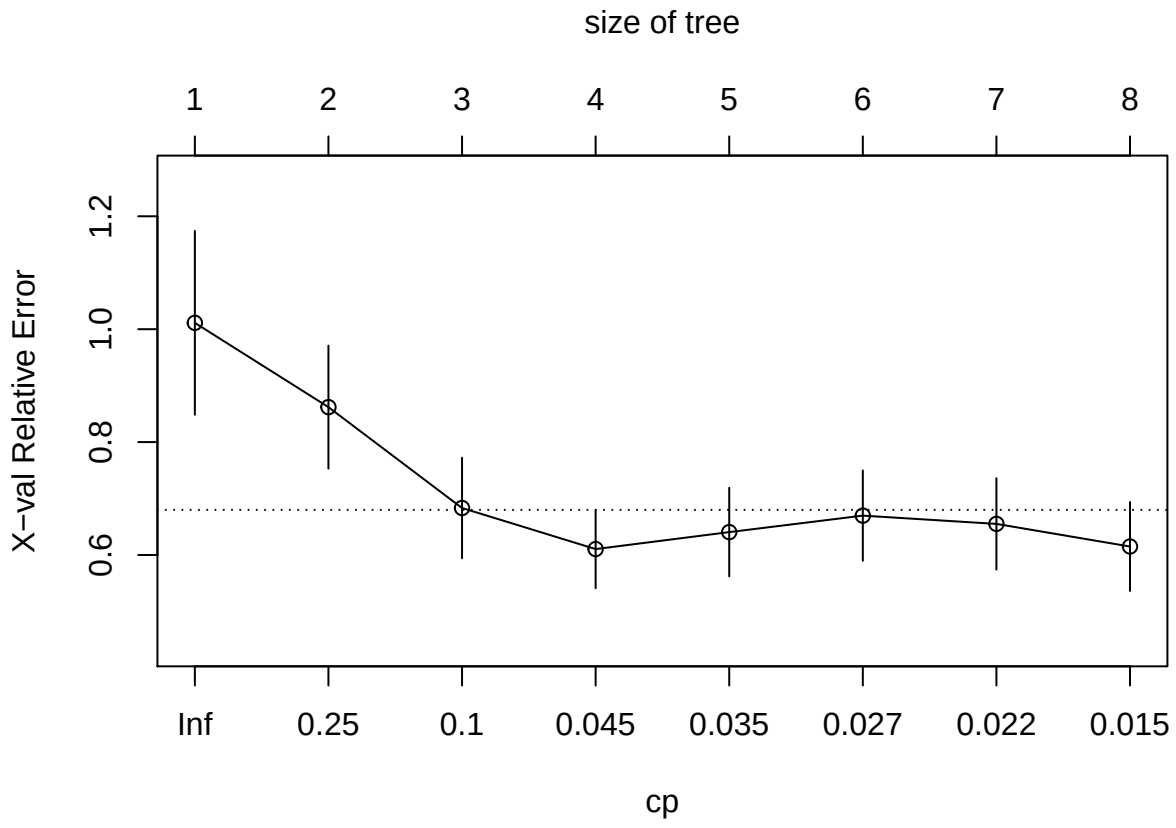



```
#print(rpart.plot(tree0))

#Tree Pruning
cpTable <- printcp(tree0)
```

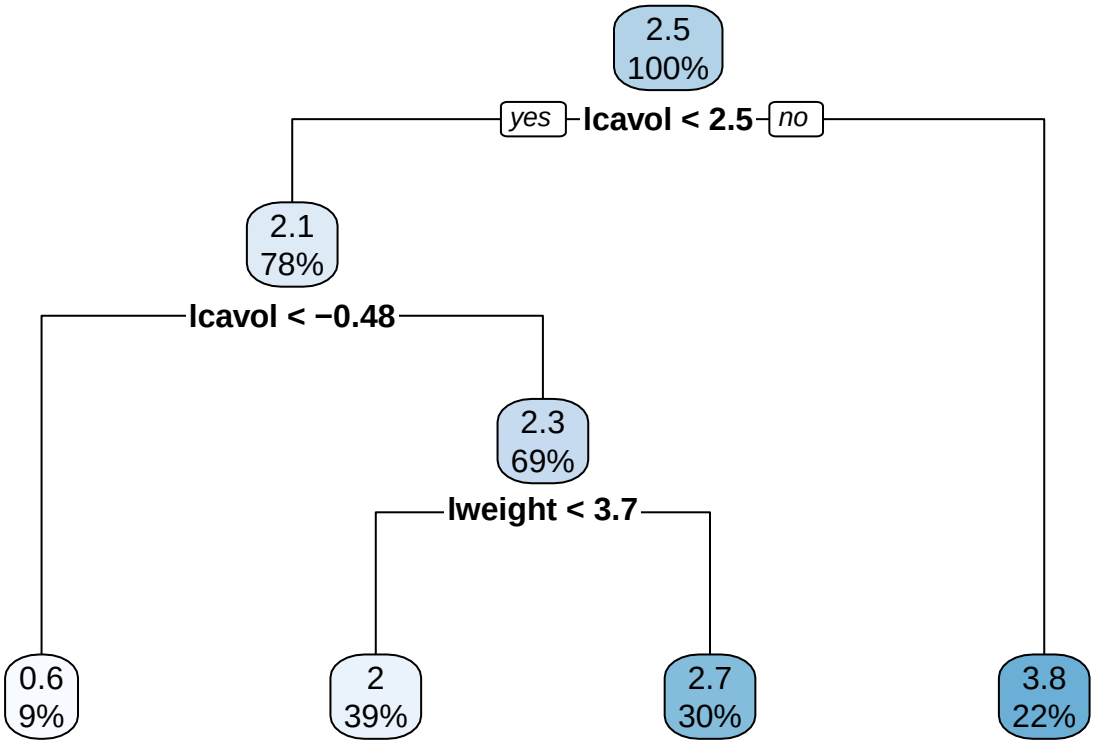
```
##
## Regression tree:
## rpart(formula = lpsa ~ ., data = prostate)
##
## Variables actually used in tree construction:
## [1] lcavol  lweight pgg45
##
```

```
## Root node error: 127.92/97 = 1.3187
##
## n= 97
##
##         CP nsplit rel error  xerror     xstd
## 1 0.347108      0   1.00000 1.01122 0.162775
## 2 0.184647      1   0.65289 0.86195 0.108934
## 3 0.059316      2   0.46824 0.68323 0.089108
## 4 0.034756      3   0.40893 0.61052 0.069407
## 5 0.034609      4   0.37417 0.64057 0.078406
## 6 0.021564      5   0.33956 0.66975 0.079990
## 7 0.021470      6   0.31800 0.65510 0.080913
## 8 0.010000      7   0.29653 0.61505 0.078746
```

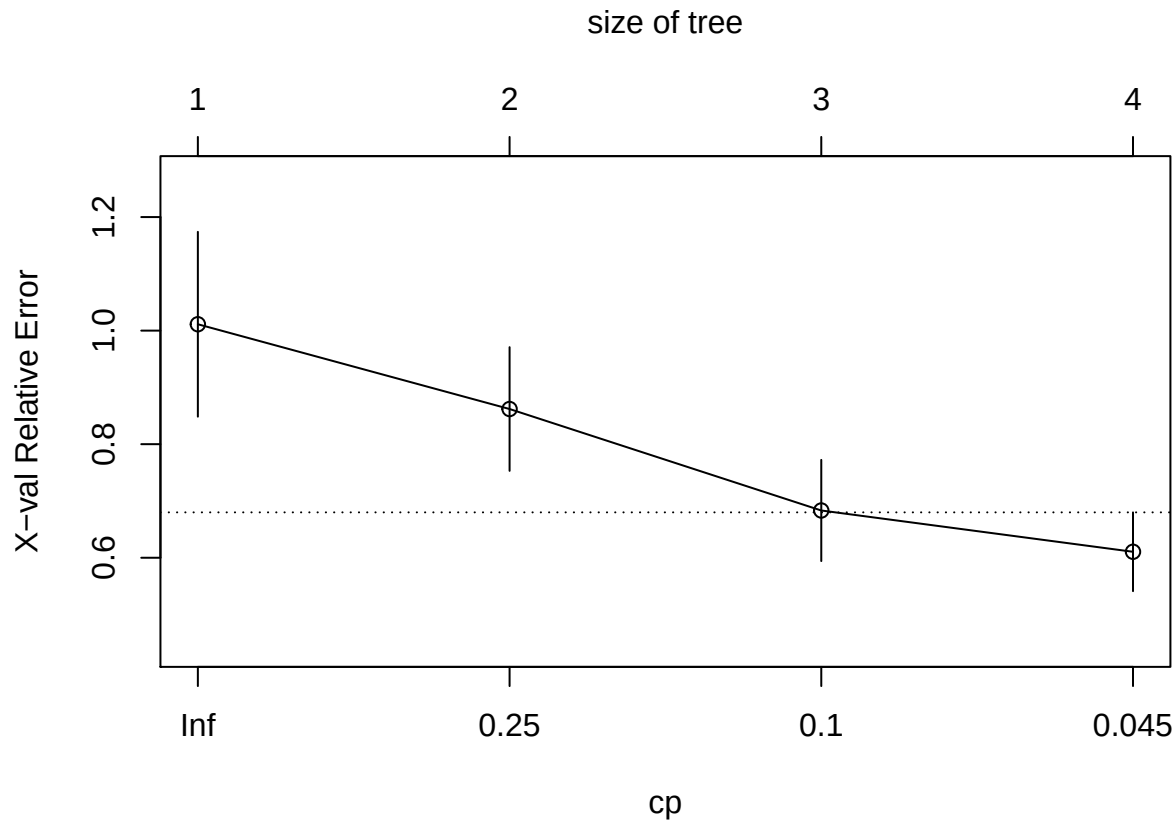
```
plotcp(tree0)
```




```
minErr <- which.min(cpTable[,4])

#The complexity parameter with the minimum cross validation error is 0.045, with a size of 4.
tree1 <- prune(tree0, cp = cpTable[minErr,1])
rpart.plot(tree1)
```

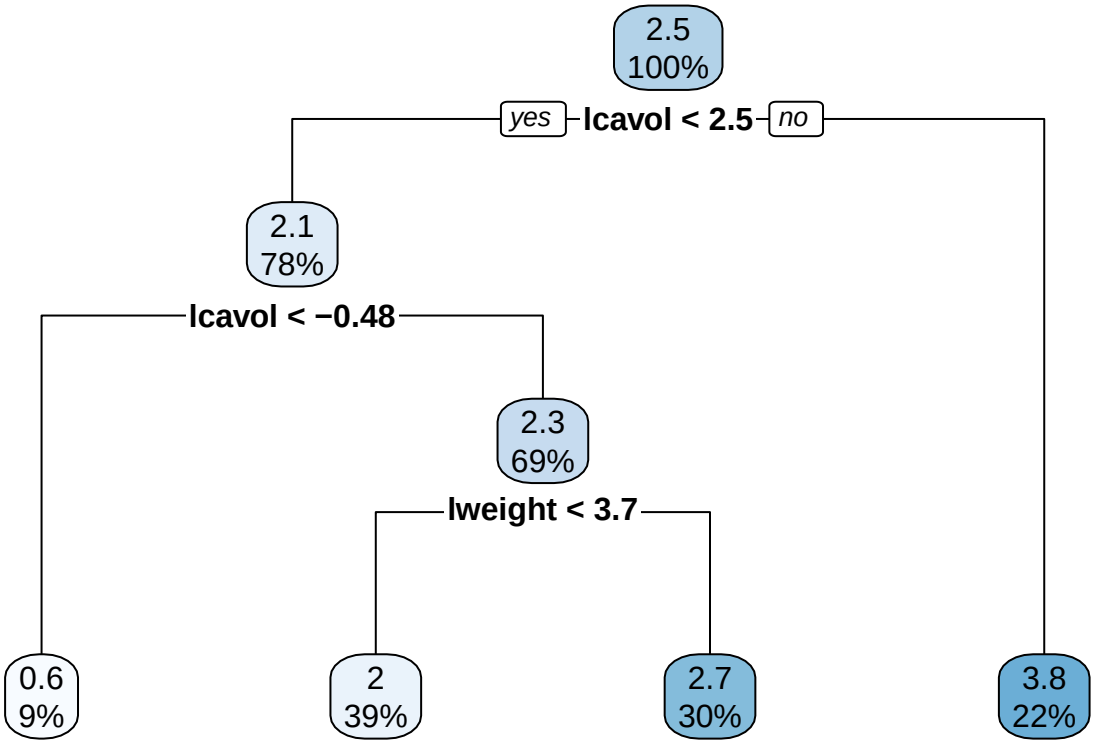



```
plotcp(tree1)
```




```
#1 SE rule
tree2 <- prune(tree0, cp= cpTable[cpTable[,4]<cpTable[minErr,4]+cpTable[minErr,5],1][1])
```

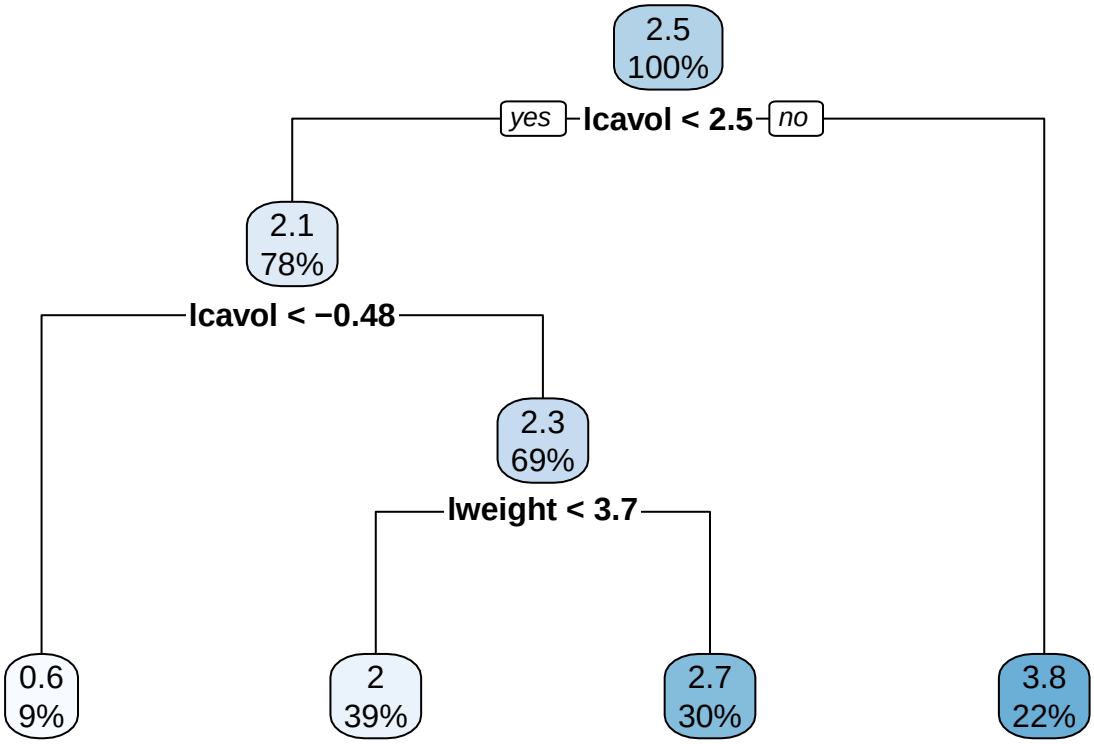
```
rpart.plot(tree2)
```



The complexity parameter with the minimum cross validation error is 0.045, with a size of 4.

This corresponds to the one standard error rule, which also has a size of 4.

Problem 1b: Tree Plot



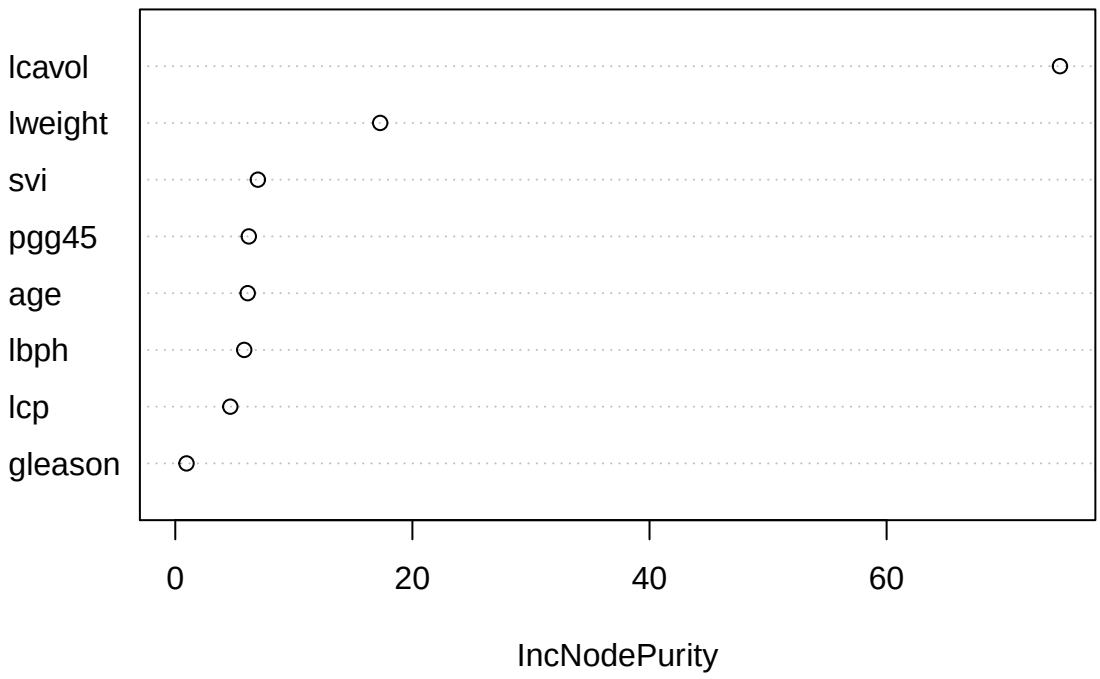
The predicted log PSA-antigen levels for a subject who has an lca volume greater than 2.5 is 3.8. 22% of observations were contained in this node.

Problem 1c: Bagging and Variable Importance

```
set.seed(2)
bagging <- randomForest(lpsa~., prostate, mtry = 8)
```

Variable Importance

```
varImpPlot(bagging)
```




```
randomForest::importance(bagging)
```

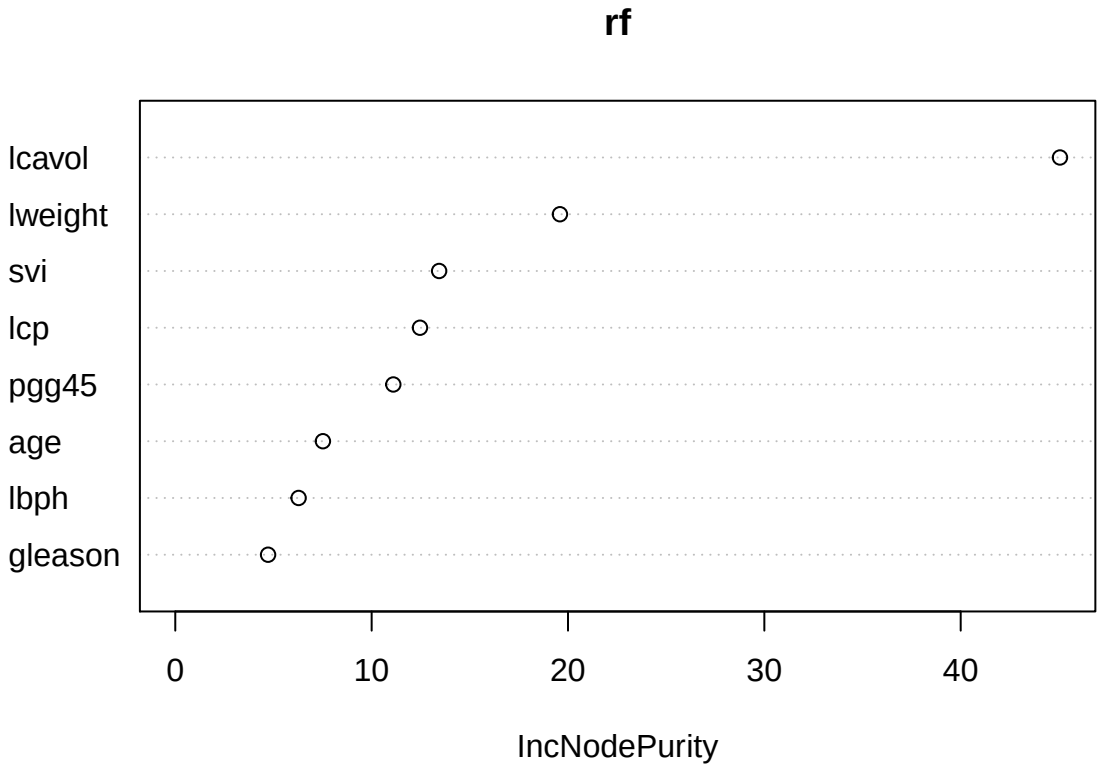
```
##         IncNodePurity
## lcavol     74.6224674
## lweight    17.2827568
## age         6.1091078
## lbph        5.8125416
## svi         6.9647589
## lcp         4.6410581
## gleason     0.9471151
## pgg45       6.2012519
```

Problem 1d: Random Forest and Variable Importance

```
set.seed(2)
rf <- randomForest(lpsa~., prostate, mtry = 3)
```

Variable Importance

```
varImpPlot(rf)
```

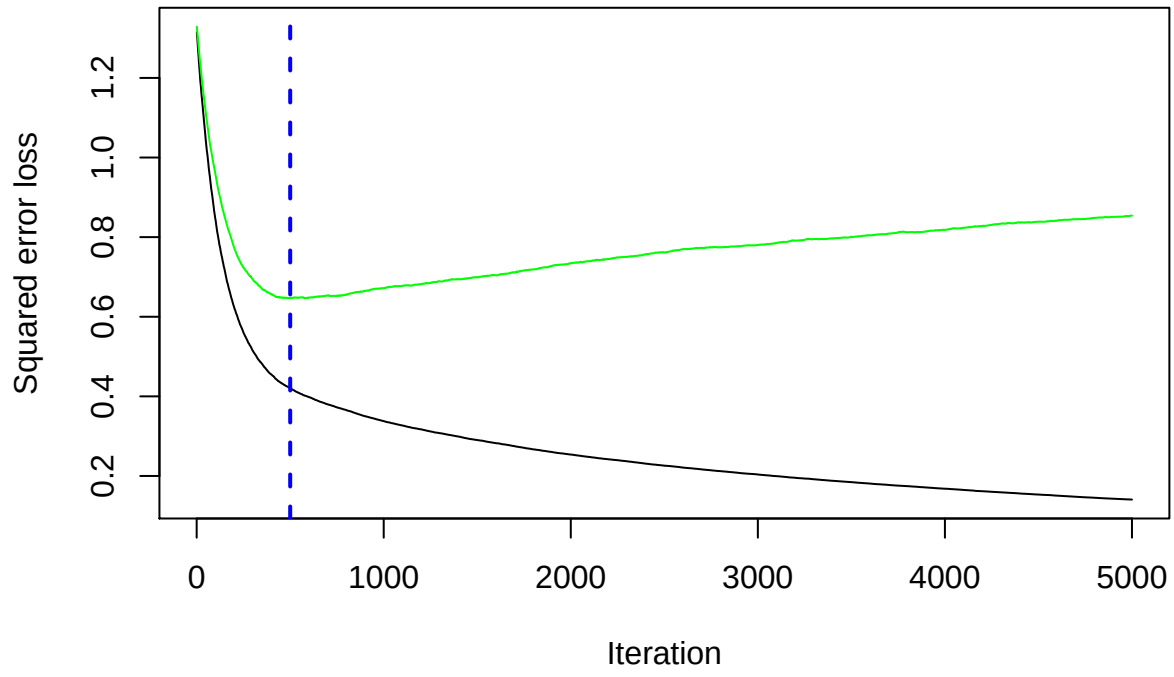



```
randomForest::importance(rf)
```

```
##         IncNodePurity
## lcavol      45.056421
## lweight     19.589659
## age          7.518539
## lbph         6.280990
## svi         13.437996
## lcp         12.461971
## gleason      4.725458
## pgg45       11.103954
```

Problem 1e: Boosting and Variable Importance

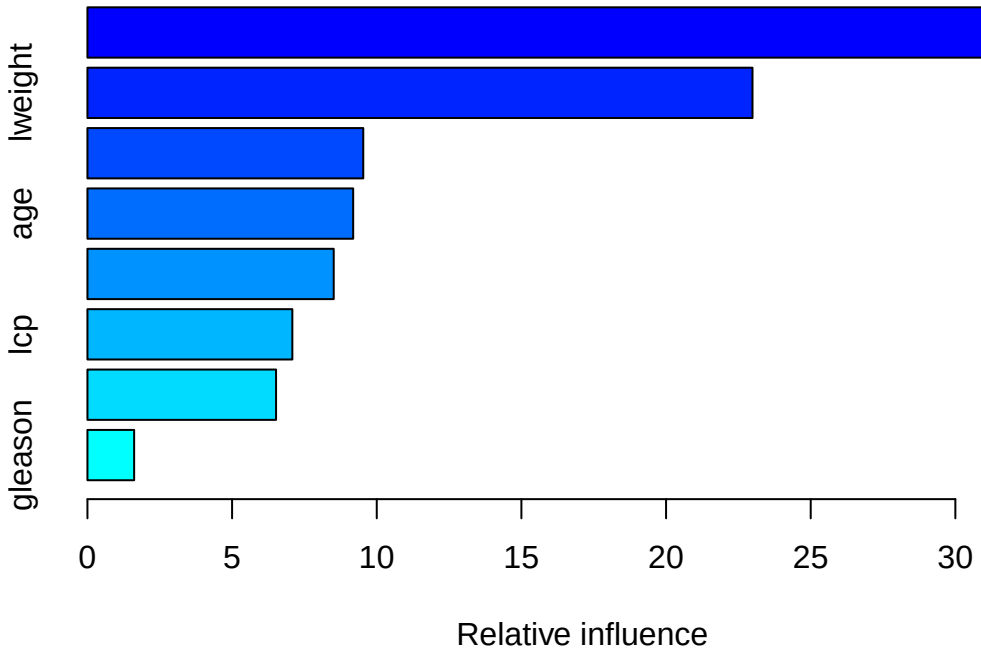
```
set.seed(2)
bst <- gbm(lpsa~., prostate, distribution = "gaussian",
           n.trees = 5000,
           interaction.depth = 3,
           shrinkage = 0.005,
           cv.folds = 10)

nt <- gbm.perf(bst, method = "cv")
```



The optimal number of trees is 573.

Variable Importance

```
summary(bst)
```




```
##             var   rel.inf
## lcavol   lcavol 34.563255
## lweight lweight 22.987817
## lbph       lbph  9.533448
## age         age  9.185766
```

```
## pgg45     pgg45  8.512590
## lcp         lcp  7.083221
## svi         svi  6.520010
## gleason gleason  1.613893
```

Problem 1f: Model Selection

To compare models, we are going to summarize the cross-validation error.

```
summary(bagging$mse)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6149  0.6187  0.6210  0.6324  0.6266  1.0906
```

```
summary(rf$mse)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6043  0.6151  0.6195  0.6298  0.6242  1.4900
```

```
summary(bst$cv.error)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6463  0.7021  0.7730  0.7663  0.8160  1.3287
```

-Cross Validation Error for Regression Tree is 0.6105232

-Cross Validation Error for Boosting is 0.6319

-Cross Validation Error for Bagging is 0.6149

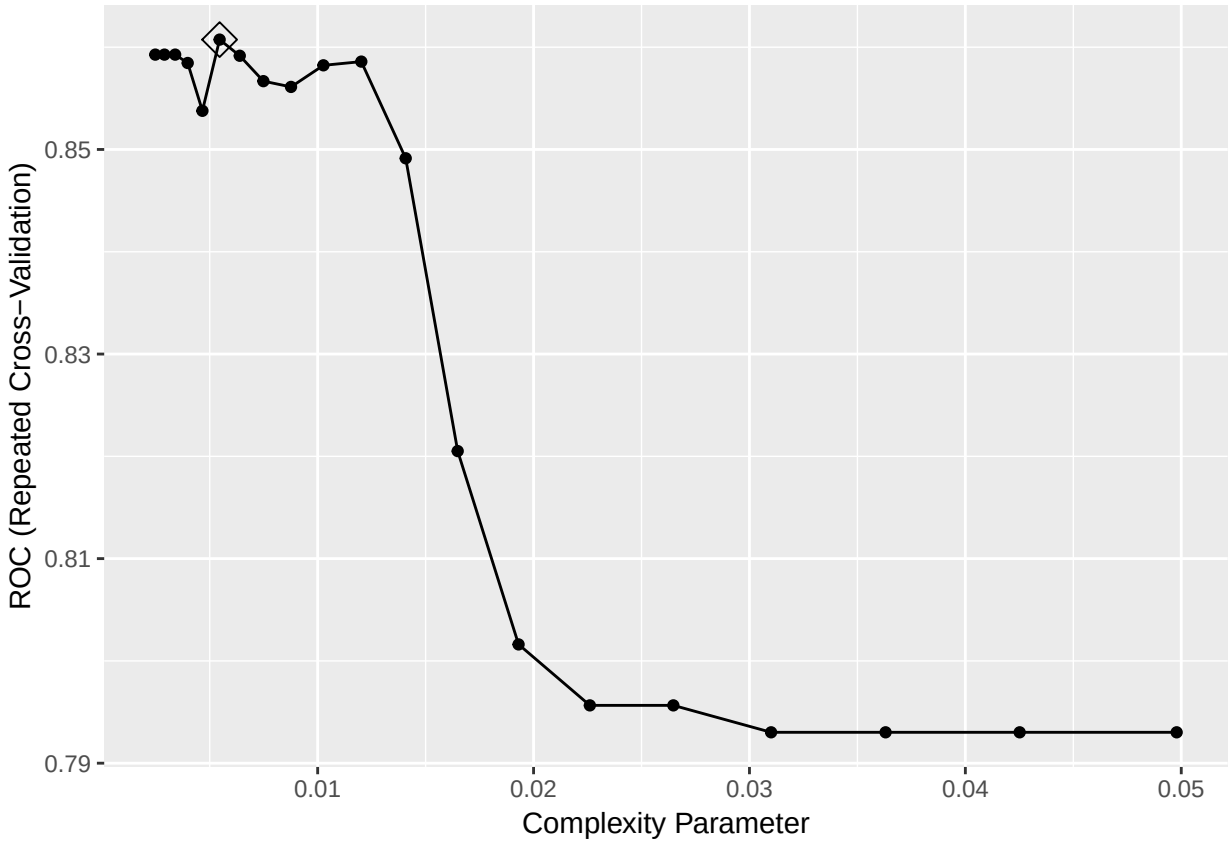
-Cross Validation Error for Random Forests is 0.6043

Random forests is the best model, with the lowest cross validation error.

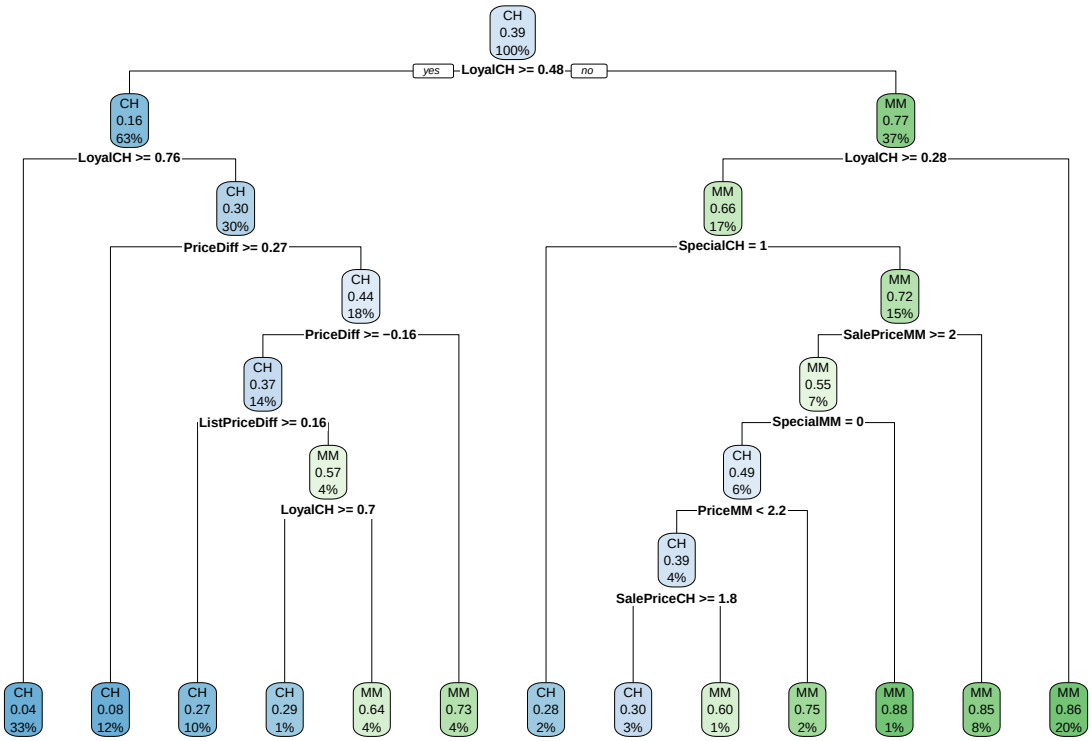
Problem 2a:

```
#Creating Partition
data(OJ)
oj.data <-OJ
n = 799/1070
rowTrain <- createDataPartition(y = oj.data$Purchase, p=n, list = FALSE)

training <-oj.data[rowTrain,]
testing <-oj.data[-rowTrain,]

#Fitting Classification Tree
ctrl <- trainControl(method = "repeatedcv", summaryFunction = twoClassSummary, classProbs = TRUE)

set.seed(2)
rpart.fit <- train(Purchase~., oj.data, subset = rowTrain,
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-6,-3, len = 20))),
                   trControl = ctrl,
                   metric = "ROC")
```

```
#Tree size is 9

ggplot(rpart.fit, highlight = TRUE)
```




```
rpart.plot(rpart.fit$finalModel)
```




```
print(rpart.fit$bestTune)
```

```
##          cp
## 6 0.0054588
```

```
#The complexity parameter is 0.0054588

rf.pred <- predict(rpart.fit, newdata = oj.data[-rowTrain,])

#Confusion Matrix
table(rf.pred, testing$Purchase)
```

```
##
## rf.pred  CH  MM
##      CH 137  22
##      MM  28  83
```

```
mean(rf.pred==testing$Purchase)
```

```
## [1] 0.8148148
```

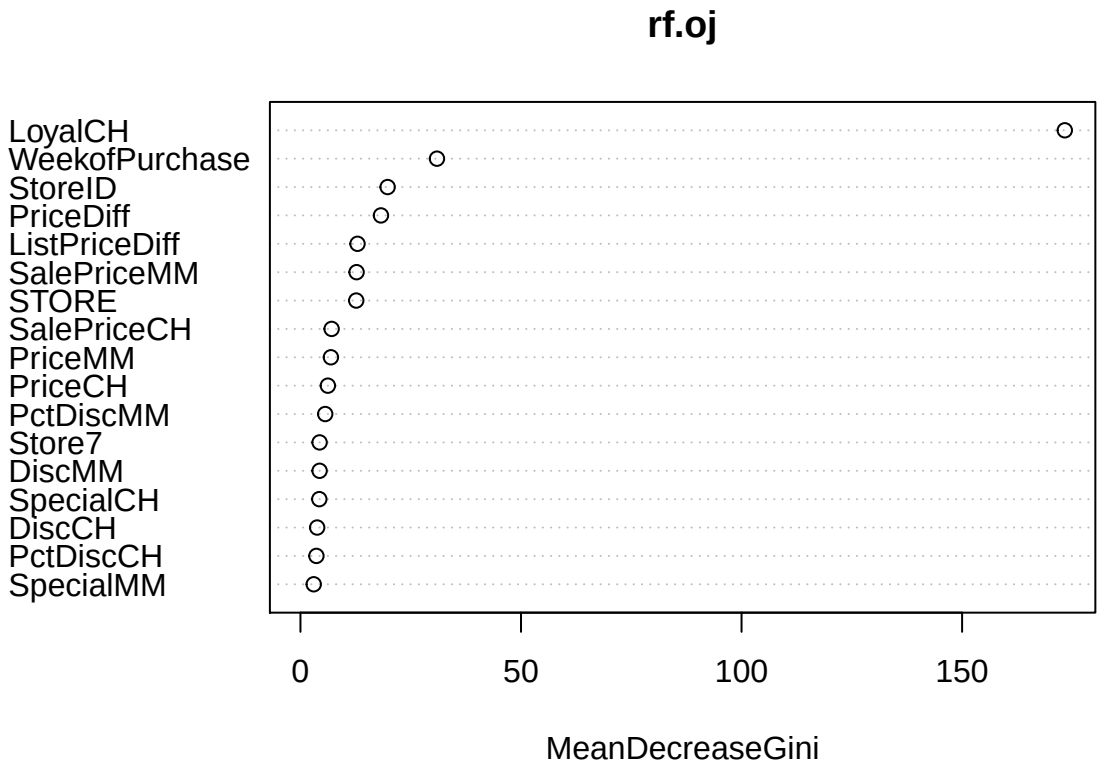
0.8148148 is the test classifcation rate

Problem 2b:

```
set.seed(2)
rf.oj <- randomForest(Purchase~., oj.data[rowTrain,], mtry = 5)
```

Variable Importance

```
varImpPlot(rf.oj)
```




```
randomForest::importance(rf)
```

```
##         IncNodePurity
## lcavol      45.056421
## lweight     19.589659
## age          7.518539
## lbph         6.280990
## svi         13.437996
## lcp         12.461971
## gleason      4.725458
## pgg45       11.103954
```

Test Error Rate

```
set.seed(2)
rfoj.pred <- predict(rf.oj, oj.data[-rowTrain,])

#Confusion Matrix
table(rfoj.pred, testing$Purchase)
```

```
##
## rfoj.pred  CH  MM
##        CH 137  29
##        MM  28  76
```

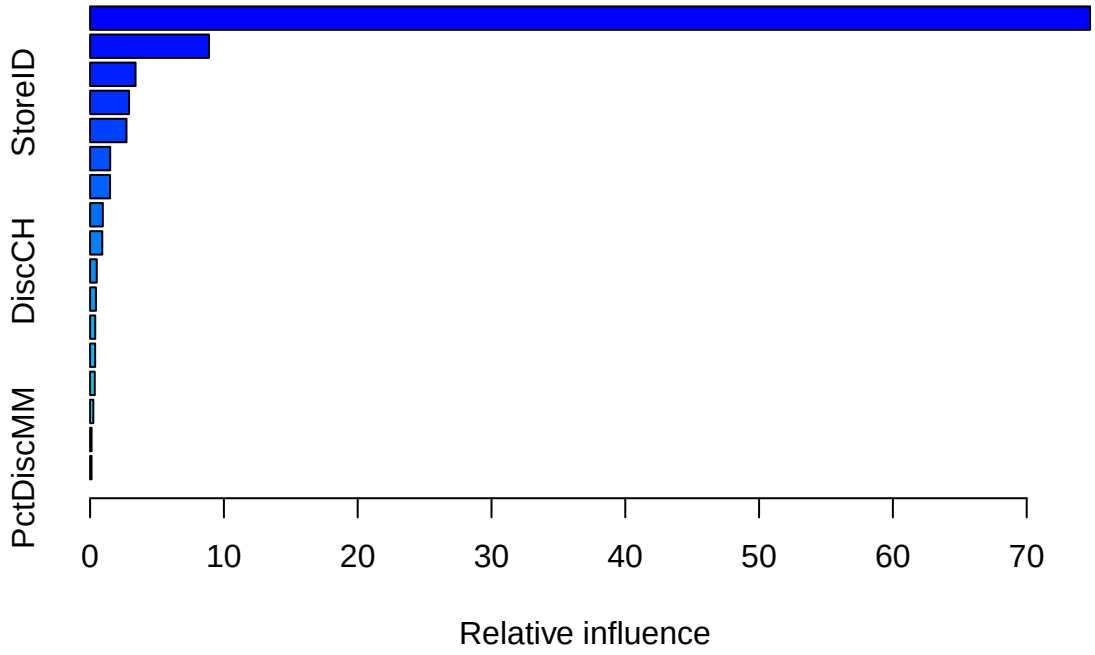
```
mean(rfoj.pred==testing$Purchase)
```

```
## [1] 0.7888889
```

0.7888889 is the test error classifcation rate

Problem 2c:

```
set.seed(2)

gbmB.grid <- expand.grid(n.trees = c(2000,3000,4000),
                         interaction.depth = 1:6,
                         shrinkage = c(0.001,0.003,0.005),
                         n.minobsinnode = 1)
# Binomial loss function
bst.oj <- train(Purchase~., oj.data,
                  subset = rowTrain,
                  tuneGrid = gbmB.grid,
                  trControl = ctrl,
                  method = "gbm",
                  distribution = "bernoulli",
                  metric = "ROC", verbose = FALSE)



summary(bst.oj)
```




```
##                           var    rel.inf
## LoyalCH               LoyalCH 74.7363140
## PriceDiff           PriceDiff  8.8870295
## ListPriceDiff   ListPriceDiff  3.3972795
```

```
## StoreID               StoreID  2.9095292
## SalePriceMM       SalePriceMM  2.7177053
## WeekofPurchase WeekofPurchase  1.5017364
## SpecialCH           SpecialCH  1.4904784
## STORE                   STORE  0.9500555
## PriceMM               PriceMM  0.9093552
## DiscCH                 DiscCH  0.4987553
## SalePriceCH       SalePriceCH  0.4354564
## SpecialMM           SpecialMM  0.3847753
## PriceCH               PriceCH  0.3770394
## DiscMM                 DiscMM  0.3463170
## Store7Yes           Store7Yes  0.2403318
## PctDiscCH           PctDiscCH  0.1094859
## PctDiscMM           PctDiscMM  0.1083558
```

```
set.seed(2)
bstoj.pred <- predict(bst.oj, oj.data[-rowTrain,])
table(bstoj.pred, testing$Purchase)
```

```
##
## bstoj.pred  CH  MM
##         CH 144  23
##         MM  21  82
```

```
mean(bstoj.pred==testing$Purchase)
```

```
## [1] 0.837037
```

0.837037 is the test error classifcation rate